REMARKS

Claims 1-18 are presently pending in this application. Claims 15-18 have been amended to more particularly define the invention.

It is noted that the claim amendments are made only for more particularly pointing out the invention, and not for distinguishing the invention over the prior art, narrowing the claims or for any statutory requirements of patentability. Further, Applicant specifically states that no amendment to any claim herein should be construed as a disclaimer of any interest in or right to an equivalent of any element or feature of the amended claim.

Claims 15-18 were rejected under 35 U.S.C. §112, second paragraph, with the contention that the word "those" resulted in the claims being unclear. By the above amendments, the word "those" has been deleted, thus overcoming this rejection.

Claims 1-7, 9, and 11-16 were rejected under 35 U.S.C. §103(a) as being unpatentable over Sreedhar, et al., U.S. Patent No. 6,182,284 B1 in view of Sastry, United States Patent Publication No. 2002/0166115 A1. Dependent claims 8, 10, 17 and 18 were rejected under 35 U.S.C. §103(a) as being unpatentable over Sreedhar in view of Sastry, and further in view of Van Dyke, United States Patent No. 5,175,856. These rejections are respectfully traversed.

In one aspect, the claimed invention is directed to a method for determining the correctness of a potential interprocedural dead store optimization for an optimizing compiler. In another aspect, the claimed invention is directed to a computer program product for the compilation of computer code. In a further aspect, the claimed invention is directed to an optimizing compiler. In yet another aspect, the claimed invention is directed to a component for determining the correctness of a potential interprocedural dead store optimization for an

optimizing compiler.

Attached are definitions of "call graph" and "control flow graph" which were obtained from readily accessible Internet sites. A call graph is a diagram that identifies modules in a system or computer program and shows which modules call one another. These modules are procedures. Thus, a call graph shows the relationship between procedures; i.e., INTERprocedural relationships. In contrast, a control flow graph is a diagrammatic representation of the possible alternative control flow paths through a component, or an abstract representation of a procedure. Thus, a control flow graph shows the relationships within a procedure; i.e., INTRAprocedural relationships.

The claimed invention relates to INTERprocedural optimization. Each aspect of the claimed invention is directed to interprocedural dead store optimization, which deals with analysis ACROSS procedures. The invention involves a call graph, which shows the relationship between procedures. The invention involves determining a live on exit set of variables for procedures, and determining a live on exit set of variables for procedure call points. Thus, these are global variables.

Sreedhar relates to INTRAprocedural optimization, which deals with analysis WITHIN a procedure. Sreedhar analyzes sets of variables that are live at the beginning and end of basic blocks. See Sreedhar at column 2, lines 57-63. Thus, these are variables within a procedure. Thus, Sreedhar is concerned with control flow graphs and local variables.

Sastry is likewise concerned with control flow graphs. See Sastry at page 2, paragraph 0012. Thus, Sastry likewise relates to INTRAprocedural optimization.

The Office Action quotes from Van Dyke "The symbol node 110 points to an array of

bit vectors 150, each bit vector containing one entry for each block node 106 in the program. This array is indexed by the depth-first numbering of the block nodes 106." Block nodes are within procedures.

A person skilled in the art, after reviewing Sreedhar, Sastry, and Van Dyke, would not find it obvious to go from live on exit from basic blocks, as in Sreedhar, to live on exit from procedures, as in the claimed invention. Sreedhar's INTRAprocedural optimization requires checking only within a single procedure for a local variable. The claimed invention involves INTERprocedural optimization, and so involves global variables. All locations within the program from which the procedure is called must be determined. Further, this must be done in a top down fashion. All call points must be checked.

Sreedhar, Sastry, and Van Dyke each relate to intraprocedural optimization. Thus, even if combined, they would still relate only to intraprocedural optimization, and so would not suggest the claimed invention.

It is accordingly submitted that the claims distinguish from the references in an unobvious manner.

In view of the foregoing, Applicant submits that claims 1-18, all the claims presently pending in the application, are patentably distinct over the prior art of record and are allowable, and that the application is in condition for allowance.

Should the Examiner find the application to be other than in condition for allowance, the Examiner is requested to contact the undersigned attorney at the local telephone number listed below to discuss any other changes deemed necessary in a telephonic or personal interview.

Serial No. 09/965,521
Docket No. CA9-2000-0027US1
YOR.564

To the extent necessary, Applicant petitions for an extension of time under 37 CFR §1.136. Please charge any shortage in the fees due in connection with the filing of this paper, including extension of time fees, to Attorney's Deposit Account No. 50-0481 and please credit any excess fees to such deposit account.

Respectfully Submitted,

Date: August 16, 2005

James N. Dresser, Esq.
Registration No. 22,973

McGinn & Gibb, PLLC
8321 Old Courthouse Road, Suite 200
Vienna, VA 22182-3817
(703) 761-4100
Customer No. 21254

CERTIFICATE OF FACSIMILE TRANSMISSION

I hereby certify that I am filing this Amendment Under 37 C.F.R. §1.116 by facsimile with the United States Patent and Trademark Office to Examiner Satish Rampuria, Group Art Unit 2124 at fax number 571-273-8300 this 16th day of August 2005.

James N. Dresser
Registration No. 22973

Web Images Groups News Froogle Local more »
define:Call graph Search Advanced Search Preferences

Web

Definitions of **Call graph** on the Web:

- (IEEE) A diagram that identifies the modules in a system or computer program and shows which modules call one another. Note: The result is not necessarily the same as that shown in a structure chart. Syn: call tree, tier chart. Contrast with structure chart. See: control flow diagram, data flow diagram, data structure diagram, state diagram.
  www.validationstation.com/glossary/glossaryc.htm

define:Call graph Search

Language Tools | Search Tips | Dissatisfied? Help us improve

Google Home - Advertising Programs - Business Solutions - About Google

Web Images Groups News Froogle Local more »

define: Control flow graph Search Advanced Search Preferences

Web

Definitions of **Control flow graph** on the Web:

- The diagrammatic representation of the possible alternative control flow paths through a component.
  www.testingstandards.co.uk/living_glossary.htm
- A control flow graph (CFG) is an abstract data structure used in compilers. It is an abstract representation of a procedure or program, maintained internally by a compiler. Each node in the graph represents a basic block, i.e. a straight-line piece of code without any jumps or jump targets; jump targets start a block, and jumps end a block. Directed edges are used to represent jumps in the control flow. There are, in most presentations, two specially designated blocks: the entry block, throu
  en.wikipedia.org/wiki/Control_flow_graph

define: Control flow graph Search

Language Tools | Search Tips | Dissatisfied? Help us improve

Google Home - Advertising Programs - Business Solutions - About Google